**How to get sub list of Vector example in java**

[**JAVA COLLECTIONS**](http://beginnersbook.com/category/java-collections/)

In this example, we are gonna see how to get a sublist of elements from a Vector. We will be using subList() method of Vector class to do this.  
More about this method from [**javadoc**](http://docs.oracle.com/javase/7/docs/api/java/util/Vector.html#subList(int,%20int)):  
public List subList(int fromIndex, int toIndex): It returns a view of the portion of this List between fromIndex, inclusive, and toIndex, exclusive. (If fromIndex and toIndex are equal, the returned List is empty.) The returned List is backed by this List, so changes in the returned List are reflected in this List, and vice-versa. The returned List supports all of the optional List operations supported by this List.  
This method eliminates the need for explicit range operations (of the sort that commonly exist for arrays). Any operation that expects a List can be used as a range operation by operating on a subList view instead of a whole List. For example, the following idiom removes a range of elements from a List:  
list.subList(from, to).clear();

**Example**

import java.util.Vector;

import java.util.List;

public class SublistExample {

public static void main(String[] args) {

// Step 1: Create a Vector

Vector<String> vector = new Vector<String>();

// Step 2: Add elements

vector.add("Item1");

vector.add("Item2");

vector.add("Item3");

vector.add("Item4");

vector.add("Item5");

vector.add("Item6");

/\* The method subList(int fromIndex, int toIndex)

\* returns a List having elements of Vector

\* starting from index fromIndex

\* to (toIndex - 1).

\*/

List subList = vector.subList(2,5);

System.out.println("Sub list elements :");

for(int i=0; i < subList.size() ; i++){

System.out.println(subList.get(i));

}

}

}

**Output:**

Sub list elements :

Item3

Item4

Item5